

SERIAL COMMUNICATIONS
OPERATION MANUAL

CAI Instruments, Inc.
152 E. Saginaw Rd.
Sanford, MI 48657
(517)6877343

INTRODUCTION

The CAI Serial Communications firmware is a group of programs contained on a EPROM (Erasable Programmable Read Only Memory) IC chip which plugs into the CAI/O interface module. The programs allow the Sinclair ZX81 or Timex/Sinclair 1000 computers to communicate with other computers, printers and other equipment using the standard serial ASCII code. The Sinclair can operate as a "DUMB" CRT terminal, dump the present screen or program listing, and send and receive character strings over the serial port.

ASCII FIRMWARE INSTALLATION

The software necessary to operate the ASCII program is provided on a 2716 EPROM which plugs into the CAI/O interface. Follow the following instructions carefully to avoid damage to the EPROM.

STEP 1 IMPORTANT-Read all the steps in this procedure before installing the EPROM.

STEP 2 Turn off the printer

STEP 3 Power Down the computer.

STEP 4 Remove the CAI/O board from the computer

STEP 5 Remove the four (4) screws from around the CAI/O case.

STEP 6 Lift the cover from the CAI/O board and expose the printed circuit board. NOTE: the three twenty-four (24) pin IC sockets at the front of the CAI/O board next the the computer connector.

STEP 7 To prevent damage to the EPROM, first neutralize the charge between the CAI/O board and the EPROM by placing the EPROM and foam carrier (foam side down) on top of the CAI/O circuit board.

STEP 8 With the foam carrier still resting on the CAI/O printed circuit board, remove the EPROM from the foam carrier. NOTE the alignment mark at the top of the EPROM.

STEP 9 Insert the EPROM in the front right 24 pin IC socket with the alignment mark toward the rear of the CAI/O board. Be very careful and make sure that all the EPROM pins are located correctly, and then apply gentle pressure to the EPROM until inserted into the IC socket.

STEP 10 Reassemble the CAI/O board back into its case.

STEP 11 If you find this operation is too difficult CAI will install the EPROM for you for only the cost of shipping.

PRINTER CABLE

CAI has specially design cables that can be used to interface the CAI/O board to most serial printers. If you wish to purchase one from CAI please contact us. If however, you wish to make your own cable the following are the pin assignments for this connection. For a more complete detail description refer to your CAI/O manual.

PRINTER CONNECTOR
(DB-25)
PIN NUMBER

CAI/O BOARD CONNECTOR
(10 PIN HEADER)
PIN NUMBER

20
4
6
2
3
5
7

1
2
4
5
6
8
9

MODEM CABLE

CAI also has modem cables that will allow you to interface your CAI/O board to a modem. If you wish to purchase one simply call CAI and state the type of modem and we will be happy to send you one. If you wish to make your own cable, the following are the pin assignments for this connection. For a more complete description refer to your CAI/O manual.

MODEM CONNECTOR
PIN NUMBER

CAI/O BOARD CONNECTOR
PIN NUMBER

6
5
20
3
2
4
7

1
2
4
5
6
7
9

MODEM INSTALLATION

STEP 1 Connect the Modem cable from the CAI/O board to the Modem. Be sure that the Modem cable is connected to the CAI/O board with the flat cable going the left.

STEP 2 Plug the Modem power supply into a wall socket and into the Modem.

STEP 3 There are two switches on the front of the Modem.

- a. Orgin/Off/Answer
- b. Half/Full/Duplex

The switches are set in the following positions for operation.

- a. Orgin(ate)
- b. Full duplex

TERMINAL EMULATION PROGRAM (TEP)

INTRODUCTION The Terminal Emulation Program (TEP) allows the Sinclair or Timex 1000 computer to operate as a standard ASCII CRT terminal. Characters typed on the keyboard are translated to ASCII code and sent over the serial port to a receiving device. Characters received by the serial port are translated to Sinclair codes and displayed on the television screen. The program allows the computer to operate in the slow mode at all times whether receiving or sending. This allows you to view the screen at all times and eliminates the flicker normally observed when the computer is operated in the fast mode.

INITIATING TERMINAL EMULATION PROGRAM The TEP is operated by simple USR function calls from BASIC. The standard terminal program sets the data rate to 300 baud, initializes the variables used by the program, and enters terminal mode.

The command LET A=USR 13312 will activate the standard terminal program. This sets the computer to transmit all information entered from the keyboard to the serial port at 300 baud. This is the main command that will be used when you want to communicate to the computer networking systems in the FULL DUPLEX MODE. The computer can be returned to the BASIC mode by entering a CONTROL 0 (SHIFT FUNCTION 0) from the keyboard (The control function is explained in a later section).

LOCAL ECHO IF the device to which the Sinclair or Timex 1000 is connected does not echo characters, you will need to set the TEP for local echo. This is set by entering the command POKE 16417,1 from the keyboard. This command will set a flag to cause characters to be displayed on the screen as they are sent. This mode is called "HALF-DUPLEX" operation. If the computer is interfaced to a serial printer and you are in this mode, the information will be displayed on the television screen and sent directly to the printer. If you wish to return to the "FULL DUPLEX" mode enter the command "POKE 16417,0".

SETTING AN ALTERNATE BAUD RATE If you wish to communicate to other serial devices at a baud rate other than 300, there are other USR function calls that allow for special initialization of the serial port hardware as described in the CAI/O interface module operation manual. The first step is to set the Asynchronous Communications Element (ACE). This is programmed with the desired rate division, word length, and modem control. The following program (for example) can be entered into the computer to set the baud rate at 110. For a complete description of how to set the baud rate refer to the CAI/O manual.

10 LET ACE=14336	Address of ACE
20 POKE ACE+3,128	Enable Baud rate Select
30 POKE ACE,27	Set LSW of Baud Divisor
40 POKE ACE+1,7	Set MSW of Baud Divisor
50 POKE ACE+3,3	Set Word Length 8
60 POKE ACE+4,3	Set RTS and DTR

After this program is run you can enter the following statement:

```
LET X=USR 13315
```

This will initialize the terminal program and enter the TEP mode using the baud rate you have selected.

ALTERNATING BETWEEN BASIC AND TERMINAL MODE-The TEP program will allow you to alternate between a BASIC program and the TEP without clearing the television screen. This is done by incorporating the following command in your BASIC program:

```
LET A=USR 13318
```

This command can be used in a BASIC program to enter the terminal mode and does not clear screen or initialize the variables required by the terminal program. It can only be used if USR 13312, 13315 or 13324 have been used before it. Also, the BASIC program using this statement should not use CLS or SCROLL if you wish to return to terminal mode with USR 13318.

If you wish only to initialize the ACE to 300 baud, but not to enter terminal mode, call USR function 13321. If you wish to

only initialize the terminal program variables and clear the screen use the USR function 13324. These two user functions (13321 and 13324) are extremely useful to initialize the serial port (ACE) and program variables prior to a screen dump, listing, sending or receiving character strings.

USING THE TERMINAL EMULATOR PROGRAM

The Sinclair and Timex 1000 keyboard do not contain all the ASCII characters needed to communicate with another ASCII device. Therefore, several shifted keys are used for characters that are not printed on the keyboard. For example with the TEP uses the shift 1 key to send an exclamation mark "!". The enclosed keyboard display shows the keyboard layout (See appendix).

CONTROL FUNCTION-The Sinclair and Timex 1000 computers do not have a control function key so the program implements a "Control Mode" by use of the FUNCTION key. The control function is actuated by pressing the FUNCTION (SHIFT-ENTER) key. The number keys implement special functions if typed in the control mode and do not send any data. The alphabetic keys send the control code for the key pressed. See appendix for an explanation of ASCII codes. The control mode can be cancelled by pressing another SHIFT-ENTER key. A special cursor is displayed when in the control mode.

The non-alphabetical keys perform the following functions in the control mode:

SHIFT ENTER	Leave control mode and enter terminal mode, or leave terminal and enter control mode.
0	Leave terminal mode and return to BASIC mode.
1	Begin saving received data in character string S\$.
2	Stop saving data in S\$.
3	Erase the screen
4	Save the screen contents in S\$
9	Print the screen contents on the CAI P/40 printer.
SPACE	Send a "BREAK" for 200 ms.

DISPLAY OF NON-SINCLAIR CHARACTERS ON INPUT-Since the Sinclair and Timex 1000 computers do not include several of the shifted

symbols, they are displayed as the inverse symbol of the key used in conjunction with the SHIFT key to generate the character. For example when an exclamation point is received by the TEP it will be displayed on the television screen as an inverse 1 (See TEP Keyboard Chart).

STOPPING AND STARTING OUTPUT-The SHIFT S and SHIFT Q send Control S and Control Q characters to the serial port. These control functions are used by many host computers to stop and start an output, respectively.

CONTROL CODES-The following control codes are recognized when received by the computer. All others are ignored.

- CR-Carriage Return
- LF-Line Feed
- FF-Form Feed/Clear Screen
- HT-Tab
- BS-Back Space

LOWER CASE CHARACTERS-The lower case letters cannot be sent from the TEP. Lower case letters being received by the TEP will be displayed as upper case.

SAVING DATA AS IT IS RECEIVED-Characters can be saved by the TEP as they are received and displayed on the television screen. Control-1 turns on the character saving mode and Control-2 turns it off. The saving mode can be turned on and off as often as necessary. Whenever saving is turned on characters are appended to the string S\$ as they are received. If you leave the TEP (return to BASIC) and then re-enter the TEP saving will start over with the first character position in S\$. The string is over written but not cleared by the TEP.

The string S\$ must be dimensioned before attempting to save characters. Saving will automatically stop when S\$ is full. The string S\$ is dimensioned before entering the TEP. For example if you enter the statement DIM S\$(100) and then enter the TEP you will have reserved space for 100 characters to be saved.

NOTE: The run command clears all variables, so run programs using the GOTO if you do not want to lose data in a string.

SAVING THE ENTIRE SCREEN The information on the entire Television screen can be saved in the string S\$. The string must be dimensioned to hold at least 768 characters in order to save the entire screen. If S\$ is not dimensioned large enough, saving will stop when S\$ is filled. Control-4 causes the the screen contents to be appended to the string S\$ overwriting but not clearing data which may have been in S\$ when TEP was entered. If S\$ is dimensioned large enough, several screens can be saved into memory. Screen saving and saving characters may be used together.

PRINTING THE SCREEN The entire screen can be printed on a CAI/P40 printer without leaving the TEP mode. Control-9 prints the screen to the printer. Characters cannot be sent or received while printing.

SCREEN DUMP PROGRAM (SDP)

This SDP is disigned to dump the contents of the television screen to the serial port. The program is designed to be used with a serial printer. Since the screen is cleared before any command entered from the keyboard is executed, the Screen Dump Program must be called from within a BASIC program to be useful. The command LET A=USR 13327 will cause whatever is on the television screen to be sent to the serial port (example in Appendix). The contents of the screen are translated from Sinclair or Timex characters to ASCII characters. Only standard video (not inverse) characters, numbers, and symbols can be sent. Graphic characters are ignored. Inverse characters are sent as the non-inverse counterpart.

The baud rate must be set before the SCREEN DUMP PROGRAM is called. This is done by entering LET A=USR 13321 to set the baud rate at 300 and then entering the command LET A=USR 13327. If you wish to use another baud rate other than 300 it can be set as outlined in the TERMINAL EMULATOR PROGRAM (TEP) section under Aternate Baude Rate Selection.

PROGRAM LISTING DUMP PROGRAM (PLDP)

This program is designed to send the current BASIC program stored in memory over the serial port. This program can be used to send a program listing to another computer or to a standard serial ASCII printer. The screen is cleared by the list program and the program lines being sent to the serial port are displayed at the top of the screen as they are sent. If the listing is sent in the slow mode you can watch the progress on the screen. In the fast mode the screen will go blank until the listing is finished.

ACCESSING THE LISTING DUMP PROGRAM-The PLDP is initiated by the user command LET A=USR 13330. This command will start the program listing. Before starting the listing you must first set the baud rate. This is achieved by entering the command LET A=USER 13321. This will set the baud rate to 300. If you wish to set the baud rate to another rate refer to the TERMINAL EMULATOR PROGRAM (TEP).

SENDING AND RECEIVING CHARACTER STRINGS (SRCS)

The SRCS function contains three USR functions that allow a BASIC program to send a character string, receive a character string, or send a character string and receive a response string through the serial port. The strings are sent and received with no translation thus allowing any type of control characters to be sent or received.

INITIALIZING THE STRINGS—The character string which is sent and received must be called S\$ and must be dimensioned. For example if you wish to send 100 characters you would dimension S\$ to 100 [DIM S\$ (100)]. This sets the space for 100 characters. The only limit on the size of S\$ is the memory space available. You can make S\$ larger than the amount of information being sent, but not smaller.

The string T\$ is used to specify a terminating character for the string read command. T\$ must not be dimensioned, but may be any length. The first character of the string is compared to incoming data from the serial port. When a match is found, the read stops. The terminating character will be placed in S\$.

SENDING A CHARACTER STRING

A character string is sent by doing the following:

1. Set the baud rate as described in the TEP.
2. Dimension S\$
3. Place the information to be sent in S\$.
4. Place CHR\$ 255 in S\$ if you do not want to send all the characters for which S\$ was dimensioned.
5. Call USR function 13333.

NOTE: CHR\$ 255 is not sent. It only marks the end of the string.

RECEIVING A CHARACTER STRING

A character string may be received when the following has been done:

1. Set the baud rate as described in TEP.
2. Dimension S\$
3. LET T\$=terminating character
4. Call USR function 13339

The read will end when S\$ is full or when a character is received which matches the first character in T\$. If the read terminates due to a match T\$(1) will appear in S\$.

ENTRY POINT ADDRESSES

- | | |
|-------|--|
| 13312 | 1) Set data speed to 300 baud.
2) Clear screen and initialize terminal program variables.
3) Enter Terminal Emulator Program |
| 13315 | 1) Clear screen and initialize terminal program variables.
2) Enter TEP mode with previously set baud rate. |
| 13318 | Re-enter TEP mode without clearing the screen (See notes 1 and 4). |
| 13321 | Initialize data speed to 300 baud. |
| 13324 | Clear screen and initialize serial communications program variables. |
| 13327 | Dump screen contents to the serial port (See note 1) |
| 13330 | Send a listing of the BASIC program to the serial port. (See note 1) |
| 13333 | Send character string S\$ to the serial port. (See notes 1 and 2) |
| 13336 | Send the string S\$ and wait for response which will be received in S\$ until T\$ is received or S\$ is filled. (See notes 1,2,3) |
| 13339 | Read string S\$ until terminator in T\$ or string is filled. (See notes 1,2,3) |

POKE 16417,1 Sets the local echo (Half Duplex)

POKE 16417,0 Sets no local echo (Full Duplex)

Note 1: 13312,13315, or 13324 must be run first if NEW, CLS, or SCROLL have been executed or the power turned off since the last time the serial communications programs were initialized.

Note 2: Character string S\$ must be dimensioned using a single dimension such as DIM S\$(100) to allocate space. Output stops when the end is reached or character CHR\$

255 is reached.

Note 3: Character string T\$ must not be dimensioned. The first character in T\$ is compared with incoming data, when a match is found or when S\$ is filled the read terminates.

Note 4: Can be only in a BASIC program.

CODE DEC (1)	HEX CODE (2)	SINCLAIR CHARACTER (3)	ASCII SENT (4)	CODE FOR ASCII SENT (5)	ASCII CHARA (6)	SINCLAIR DISPLAYED (7)	HEX CODE SINCLAIR (8)
0	00	SPACE	SPACE	20	NULL	--	--
1	01		--	--	SOH(1A)	--	--
2	02		--	--	STX(1B)	--	--
3	03		--	--	ETX(1C)	--	--
4	04		--	--	EOT(1D)	--	--
5	05		--	--	ENQ(1E)	--	--
6	06		--	--	ACK(1F)	--	--
7	07		--	--	BEL(1G)	--	--
8	08		--	--	BS (1H)	--	--
9	09		--	--	HT (1I)	--	--
10	0A		--	--	LF (1J)	--	--
11	0B		--	--	VT (1K)	--	--
12	0C		HT(TAB)	09	FF (1L)	--	--
13	0D	\$	\$	24	CR (1M)	--	--
14	0E	:	:	3A	SO (1N)	--	--
15	0F	?	?	3F	SI (1O)	--	--
16	10	(	(	28	DEL(1P)	--	--
17	11	)	)	29	DC1(1Q)	--	--
18	12	>	>	3E	DC2(1R)	--	--
19	13	<	<	3C	DC3(1S)	--	--
20	14	=	=	3D	DC4(1T)	--	--
21	15	+	+	2B	NAK(1U)	--	--
22	16	-	-	2D	SYN(1V)	--	--
23	17	*	*	2A	ETB(1W)	--	--
24	18	/	/	2F	UAN(1X)	--	--
25	19	;	;	3B	EM (1Y)	--	--
26	1A	,	,	2C	SUB(1Z)	--	--
27	1B	.	.	2E	ESC	--	--
28	1C	0	0	30	FS	--	--
29	1D	1	1	31	GS	--	--
30	1E	2	2	32	RS	--	--
31	1F	3	3	33	US	--	--
32	20	4	4	34	SPACE	SPACE	00
33	21	5	5	35	!	1*	9D
34	22	6	6	36	"	"	0B
35	23	7	7	37	#	3	0C
36	24	8	8	38	\$	\$	0D
37	25	9	9	39	%	4	A0
38	26	A	A	41	&	2	9E
39	27	B	B	42	'	W	BC
40	28	C	C	43	(	(	10
41	29	D	D	44	)	)	11
42	2A	E	E	45	*	*	17
43	2B	F	F	46	+	+	15
44	2C	G	G	47	,	,	1A
45	2D	H	H	48	-	-	16
46	2E	I	I	49	.	.	1B
47	2F	J	J	4A	/	/	18
48	30	K	K	4B	0	0	1C
49	31	L	L	4C	1	1	1D
50	32	M	M	4D	2	2	1E
51	33	N	N	4E	3	3	1F
52	34	O	O	4F	4	4	20

CODE DEC	HEX CODE	SINCLAIR CHARACTER	ASCII SENT	CODE FOR ASCII SENT	ASCII CHARA	SINCLAIR DISPLAYED	HEX CODE SINCLAIR
53	35	P	P	50	5	5	21
54	36	Q -	Q -	51	6	6	22
55	37	R	R	52	7	7	23
56	38	S	S	53	8	8	24
57	39	T	T	54	9	9	25
58	3A	U	U	55	:	:	0E
59	3B	V	V	56	;	;	19
60	3C	W	W	57	<	>	13
61	3D	X	X	58	=	=	14
62	3E	Y	Y	59	>	>	12
63	3F	Z	Z	5A	?	?	0F
64	40	RND	--	--	@	A	A6
65	41	INKEY\$	--	--	A	A	26
66	42	PI	--	--	B	B	27
67	43	--	--	--	C	C	28
68	44	--	--	--	D	D	29
69	45	--	--	--	E	E	2A
70	46	--	--	--	F	F	2B
71	47	--	--	--	G	G	2C
72	48	--	--	--	H	H	2D
73	49	--	--	--	I	I	2E
74	4A	--	--	--	J	J	2F
75	4B	--	--	--	K	K	30
76	4C	--	--	--	L	L	31
77	4D	--	--	--	M	M	32
78	4E	--	--	--	N	N	33
79	4F	--	--	--	O	O	34
80	50	--	--	--	P	P	35
81	51	--	--	--	Q -	-Q	36
82	52	--	--	--	R	R	37
83	53	--	--	--	S	S	38
84	54	--	--	--	T	T	39
85	55	--	--	--	U	U	3A
86	56	--	--	--	V	V	3B
87	57	--	--	--	W	W	3C
88	58	--	--	--	X	X	3D
89	59	--	--	--	Y	Y	3E
90	5A	--	--	--	Z	Z	3F
91	5B	--	--	--	[	R	B7
92	5C	--	--	--	\	T	B9
93	5D	--	--	--	]	Y	BE
94	5E	--	--	--		7	A3
95	5F	--	--	--		8	A4
96	60	--	--	--		E	AA
97	61	--	--	--	a	A	26
98	62	--	--	--	b	B	27
99	63	--	--	--	c	C	28
100	64	--	--	--	d	D	29
101	65	--	--	--	e	E	2A
102	66	--	--	--	f	F	2B
103	67	--	--	--	g	G	2C
104	68	--	--	--	h	H	2D
105	69	--	--	--	i	I	2E
106	6A	--	--	--	j	J	2F
107	6B	--	--	--	k	K	30

CODE DEC	HEX CODE	SINCLAIR CHARACTER	ASCII SENT	CODE FOR ASCII SENT	ASCII CHARA	SINCLAIR DISPLAYED	HEX CODE SINCLAIR
108	6C	--	--	--	l	L	31
109	6D	--	--	--	m	M	32
110	6E	--	--	--	n	N	33
111	6F	--	--	--	o	O	34
112	70	--	--	--	p	P	35
113	71	--	--	--	q	Q	36
114	72	--	--	--	r	R	37
115	73	--	--	--	s	S	38
116	74	GRAPHICS	--	--	t	T	39
117	75	EDIT	--	--	u	U	3A
118	76	ENTER	CR	OD	v	V	3B
119	77	DELETE	--	--	w	W	3C
120	78	K/L MODE	--	--	x	X	3D
121	79	FUNCTION	--	--	y	Y	3E
122	7A	--	--	--	z	Z	3F
123	7B	--	--	--	{	F	AB
124	7C	--	--	--	.	G	AC
125	7D	--	--	--	}	M	AD
126	7E	--	--	--	~	D	A9
127	7F	--	--	--	DEL	O	9C

APPENDIX DESCRIPTION

Column 1-Decimal Code-the decimal code being looked up by the computer.

Column 2-Hex code-The Hex value being looked up.

Column 3-Sinclair Character-The Sinclair Character represented by the code in columns 1 or 2.

Column 4-ASCII Character Sent-The ASCII character sent by the list and dump program when the code is encountered.

Column 5-ASCII Code Sent-The Hex Code for the ASCII character in column 4.

Column 6-ASCII Character-The ASCII character represented by the code in column 1 or 2.

Column 7-Sinclair Character Displayed-The character which is displayed on the screen by the Terminal Emulator Program when the code in column 1 or 2 is received by the serial port.

Column 8-Sinclair Code Displayed-The HEX code for the characters in column 7.

UNDERScoreD CHARACTERS INDICATE INVERSE VIDEO

EXAMPLE-Using String Send and Receive to To Communicate Between Two Sinclair or Timex 1000 Computers.

START

```

10 LET T$=CHR$ 245 TERMINATE REC
20 DIM S$(80)
30 LET A=USR 13321 (Set 300 baud)
40 PRINT "ENTER S TO SEND FIRST"
50 PRINT "ENTER R RECEIVE FIRST"
60 INPUT A$
70 IF A$="S" THEN GOTO 100
80 IF A$="R" THEN GOTO 130
90 GOTO 40
100 INPUT R$
110 LET S$=(R$+T$+CHR$ 255)
120 LET A=USR 13333 TRANSMIT #
130 LET S$="" CLEAR FOR READ & STORE
140 LET A=USR 13339 READ # UNTIL TERMINATOR IN T$ OR STRING IS FILLED
150 GOTO 230 STORE INCOMING DATA IN S$
200 INPUT R$
210 LET S$=(R$+T$+CHR$ 255)
220 LET A=USR 13336 SEND S$ STRING AND WAIT FOR RESPONSE READ
230 FOR I=1 TO 80
240 IF S$(I)=T$ THEN GOTO 270 TERMINATE READ IF T$ RECEIVED
250 PRINT S$(I);
260 NEXT I
270 GOTO 200

```

TERMINATE SEQUENCE SENT

LOAD this program into two computers. Start one sending and the other receiving, and type messages back and forth.

EXAMPLE-Using Screen Dump and Program Listing

```

10 LET A=USR 13321 (Set Baud rate 300)
20 FOR I=1 TO 20
30 FOR J=1 TO I
40 PRINT J;
50 NEXT J
60 NEXT I
70 LET A=USR 13327
80 PRINT AT 0,0 "DO YOU WANT TO LIST THE PROGRAM ? (Y=YES
  N=NO)"
90 INPUT A$
100 IF A$(1)="Y" THEN LET A=USR 13330
110 PRINT "DONE"

```


ALIGNMENT NOTCH
TOWARD REAR

PLASTIC SCREWS
(6 PLACES)

I/O PORT

TAPE PORT

PRINTER PORT

RED BAND
OR ARROW

RIBBON
CABLE

PRINTER
SOCKET

TAPE
SOCKET

MODEM
SOCKET

